

Tommi Heinäjärvi

MOODLE: MODULAARINEN
OPPIMISALUSTA
Videoaineistomoduulin toteutus

Opinnäytetyö
Tietojenkäsittely


Maaliskuu 2010




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILULEHTI

 MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences		Opinnäytetyön päivämäärä 8.3.2010
Tekijä(t) Tommi Heinäjärvi		Koulutusohjelma ja suuntautuminen Tietojenkäsittelyn koulutusohjelma
Nimeke Moodle: modulaarinen oppimisalusta – videoaineistomoduulin toteutus		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli toteuttaa Mikkelin ammattikorkeakoulun käyttämään Moodle-oppimisalustaan liitettävissä oleva videoaineistomoduuli. Moduulin rakenne oli yleiseltä toimintaperiaatteeltaan yhdessä toimeksiantajan kanssa hahmoteltu. Moduulin piti mahdollistaa ammattikorkeakoulun opettajien lisäävän Moodleen web-kameran välityksellä taltioitavaa videokuvaa. Taltiointiprosessin tulisi olla niin helppokäyttöinen, että ominaisuuden käyttöönoton kynnys on riittävän matala.</p> <p>Työn toteutus pohjautui vuonna 2008 Campus IT:n projektiympäristössä suorittamaani syventävään harjoitteluun, jonka aikana perehdyin sekä Moodlen että videojulkaisun perusteisiin. Harjoittelun aikana toteutin Campus IT:lle opinnäytetyön edistymisen seurantaan käytettävän moduulin sekä tutustuin CampusTV:n käyttämiin ratkaisuihin videojulkaisussa. Työn tuloksena syntyi tavoitteiden mukainen aineistomoduuli. Kuvaus kehitysprosessista helpottaa Moodlen ominaisuuksien kehittämistä myös jatkossa, sillä toteutetun moduulin periaatteet ovat yleistettävissä myös muita käyttötarkoituksia varten kehitettäviin lisäosiin.</p>		
Asiasanat (avainsanat) oppimisympäristö, verkko-opetus, verkko-oppimateriaali, audiovisuaalinen oppimateriaali		
Sivumäärä 32	Kieli Suomi	URN URN:NBN:fi:amk-201003105570
Huomautus (huomautukset liitteistä)		
Ohjaavan opettajan nimi Arto Väättäinen		Opinnäytetyön toimeksiantaja Campus IT

DESCRIPTION

 MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences		Date of the bachelor's thesis 8 March 2010
Author(s) Tommi Heinäjärvi		Degree programme and option Business Information Technology
Name of the bachelor's thesis Moodle: a modular e-learning platform – the development of a video resource module		
Abstract <p>The goal of the bachelor's thesis was to develop a video resource module for Moodle – a popular e-learning platform used also in Mikkeli University of Applied Sciences. The basic structure of the module was designed together with the representative of Campust IT – a project environment of Mikkeli University of Applied Sciences. The module was expected to make it possible for teachers to record videos by using their webcam and to add the recorded videos to the Moodle site for students to view. The user interface of the module was required to be simple enough so that using the module would be easy.</p> <p>My study was based on practical training that I completed in the Campus IT project environment in the summer of 2008. During my training I studied the structures of both Moodle and basic video publishing. As a result I developed my first Moodle module which formed the basis for developing the video resource module. The description of the development process will make it easier for future developers to design and develop more plugins for Moodle. Many of the basic principles described in the study apply for different circumstances as well.</p>		
Subject headings, (keywords) learning environment, e-learning, virtual learning material, audio-visual learning material		
Pages 32	Language Finnish	URN URN:NBN:fi:amk-201003105570
Remarks, notes on appendices		
Tutor Arto Väättäinen		Bachelor's thesis assigned by Campus IT

SISÄLTÖ

1 JOHDANTO.....	1
2 VIRTUAALIOPISELU.....	2
2.1 Menetelmät ja tavoitteet.....	2
2.2 Virtuaaliset oppimisympäristöt.....	3
2.3 Oppimismateriaalit.....	4
3 MOODLE VIRTUAALISENA OPPIMISYMPÄRISTÖNÄ.....	6
3.1 Käytettävät materiaalit.....	6
3.2 Tekninen rakenne.....	8
4 FLASH-FORMAATTI WEB-VIDEOIDEN JULKAISUSSA.....	12
4.1 Flash-formaatin toisto.....	13
4.2 Flash-videoiden liittäminen web-sovellukseen.....	14
5 VIDEOAINEISTOMODUULIN KEHITTÄMINEN MOODLEEN.....	16
5.1 Toimintaperiaate.....	17
5.2 Tietokantarakenne.....	20
5.3 Tietokannan hallinta.....	23
5.4 Käyttöliittymä.....	24
5.5 Kieliversioiden hallinta.....	28
6 POHDINTA.....	29
LÄHTEET.....	31

1 JOHDANTO

Opinnäytetyön toimeksiantajana toimiva Campus IT on Mikkelin ammattikorkeakoulun yhteydessä toimiva projektiympäristö, jonka tarkoituksena on tarjota ammattikorkeakoulun opiskelijoille harjoittelu- ja projektitoimeksiantoja ja tätä kautta tuottaa ammattikorkeakoulun toimintoihin liittyviä sovelluksia. Yksi ensimmäisistä projektiympäristön toimeksiannoista oli Moodle-oppimisolustan toimintaperiaatteiden selvittäminen ja dokumentointi niin, että ammattikorkeakoulun on mahdollista tulevaisuudessa laajentaa ja muokata oppimisolustaa tarpeitaan paremmin vastaavaksi.

Mikkelin ammattikorkeakoulussa ensimmäinen vaihe verkko-opetuksen keskittämisessä Moodle-ympäristöön oli luopuminen muista käytössä olleista oppimisolustoista. Viimeisin otettu askel on käytössä olleen Moodle-ympäristön päivitys uusimpaan sovelluksesta julkaistuun vakaaseen versioon. Rakenteellisten muutosten yhteydessä on koettu tarpeelliseksi selvittää valitun oppimisolustan kehitysmahdollisuuksia.

Työni tarkoitus oli yhdistää verkko-opetukseen internetissä viime vuosina laajasti alaa vallannut ominaisuus eli videokuvan julkaisu. Ennen tietotekniikan alalla tapahtuneita kehitysaskeleita internetin ensisijainen tiedonvälitysmuoto oli kuvilla havainnollistettu teksti. Useat esimerkit ovat osoittaneet, että myös verkossa on tarve ja kysyntä informaation esittämiseen videomuodossa.

Työn ensisijaisena tavoitteena oli Moodleen liitettävän moduulin toteutusvaiheiden kuvaaminen. Mikkelin ammattikorkeakoulussa ei tätä ennen ollut julkaistu opinnäytetöitä, jotka dokumentoisivat ja esittelisivät Moodle-kehityksen periaatteita, minkä vuoksi työssä keskityttiin ensisijaisesti Moodle-oppimisolustaan ja toissijaisesti Flash-videojulkaisuun.

2 VIRTUAALIOPISKELU

Verkko-opiskelu on viime vuosien aikana integroitunut yhä keskeisemmäksi osaksi opettamista ja opiskelijoiden oppimisprosessia. Tekninen kehitys on ollut ensisijaisen tärkeä edellytys verkko-opetuksen yleistymiselle, mutta myöskään uudenlaisten verkko-opetuksen metodien omaksumisen tärkeyttä ei pidä väheksyä. Uusia apuvälineitä on pyrittävä käyttämään niin, että ne tukevat toimivasta ja kokonaisvaltaisesta oppimisprosessista olemassa olevien näkemysten toteutumista.

2.1 Menetelmät ja tavoitteet

Lähiopetuksen osana verkko-oppimisen tavoitteina ovat lähinnä opetuksen havainnollistaminen, opiskelijoiden aktivointi sekä tiedollisen sisällön laajentaminen. (Keränen & Penttinen 2007, 9.) Avoimessa oppimisympäristössä opiskelijan on mahdollista valita itselleen sopivimmat oppimisvälineet. Opetuksen avoimuus johtaa siihen, että opetuksen tulee joustaa oppimisprosessin mukaisesti. Tämä tarkoittaa opettajan aseman laajenemista pelkästä luennoitsijasta henkilökohtaiseksi ohjaajaksi. (Meisalo ym. 2000, 66.)

Virtuaaliopiskelu on eräs etäopiskelun muoto, joka tarjoaa opiskelijoille vapaammat opiskelumahdollisuudet kuin perinteinen opetus. Verkko-opetus eroaa lisäksi muista etäopiskelun muodoista siinä, että se mahdollistaa paremmin sekä opettajan ja opiskelijan että opiskelijoiden välisen vuorovaikutuksen. (Flinkman ym. 2006, 16.) Ajasta ja paikasta riippumattoman verkko-oppimisympäristön tarjoamat oppimismenetelmät voivat olla perinteistä lähiopetusta tehokkaampia ja syvemmälle ulottuvia. Verkko-opetuksen toimivuuden kannalta tärkeää onkin oppimisprosessin rakenteen laadukas suunnittelu. (Silander & Koli 2003, 102.)

Verkko-opetusta kehitettäessä ja otettaessa käyttöön uusia toimintatapoja on keskeistä, että uudistukset vastaavat opiskelijoiden omia opiskelukeinoihin liittyviä näkemyksiä ja toiveita. Kajaanin ammattikorkeakoulussa tehdyssä tutkimuksessa selvitettiin opiskelijoiden suhtautumista verkko-opetuksen toteutukseen. Tärkeimmäksi seikaksi verkko-opiskelun onnistumisen kannalta nousi opiskelijoiden mielissä opettajan

asiantuntemus ja aktiivinen osallistuminen ohjauksen ja palautteen annon muodossa. (Flinkman ym. 2006, 28 – 29.) Verkko-opetuksen on jo lähtökohtaisesti oltava suunniteltu verkkoympäristön ominaisuudet huomioon ottaen. Opetuksen siirtyminen verkkoon korostaa opettajan näkökulmasta oikeanlaisen oppimateriaalin välittämistä opiskelijoille sekä oppimisprosessin ohjauksen tärkeyttä. (Karevaara 2009, 14.)

Verkko-opiskelun mukanaan tuomia etuja ovat esimerkiksi oppimisprosessin syventäminen etätehtävillä ja lähiopetuksen ulkopuolisilla keskusteluilla. Verkko-opiskelu mahdollistaa lisäksi aktiivisen paneutumisen oppimisprosessin aikana ilmenneisiin ongelmiin. Verkossa oleva aineisto myös vapauttaa opiskelijan perehtymään oppimismateriaalin haluamallaan tavalla. (Kalliala 2002, 30 – 31.)

2.2 Virtuaaliset oppimisympäristöt

Verkko-opetuksen käytännön toimivuuden edistämiseksi on oppilaitosten mahdollista siirtyä käyttämään keskitetysti yhtä virtuaalista oppimisympäristöä. Vaihtoehtoja on olemassa useita aina oppilaitoksen itselleen sopiviksi kokoamista sovelluksista kaupallisiin valmisratkaisuihin. Oppilaitoksen on valikoitava omia tarkoituksiaan parhaiten vastaava kokonaisratkaisu.

Virtuaaliopiskelu voi tapahtua myös normaalissa lähiopetustilanteessa, jossa hyödynnetään esimerkiksi internetiä, tietokoneohjelmia tai muuta virtuaalista oppimismateriaalia. Virtuaalioppimisen erottaa verkko-oppimisesta opiskelijoiden ja opettajan välinen vuorovaikutus. Virtuaalioppimisen muissa muodoissa opiskelija hyödyntää oppimateriaalia itsenäisesti opiskelussaan. Verkko-opetuksen etuja ovat mahdollisuudet opintojen aikatauluttamiseen joustavasti sekä laajat materiaalivarannot. Yksi verkko-oppimisen muodoista on verkkokurssi, joka on oppimisalustan ominaisuuksia kuten tiedostojen jakamista ja keskustelualueita hyödyntäen suoritettava kurssi. Opettajalla on keskeinen rooli verkkokurssilla opiskelijoiden motivoinnissa ja ohjauksessa. Verkkokurssilla voidaan pyrkiä saavuttamaan yhteisöllisyys esimerkiksi keskustelujen, videoiden tai ryhmätyöskentelyn kautta. (Keränen & Penttinen 2007, 2 – 4.)

Opiskelualustojen kehittyminen on tapahtunut pitkälti yleisten web-tekniikoiden kehitystä seuraten. Oppimisalustojen käytön yleistymisen voidaan ajoittaa 90-luvun puoliväliin. Nykyisin ilmaisten ja kaupallisten oppimisalustojen väliset erot käytettävyydessä kiteytyvät ennen kaikkea kaupallisten palvelujen tarjoamiin lisäominaisuuksiin, joita voivat olla esimerkiksi materiaalin tuotantoon ja hallinnointiin sekä erilaisiin tukipalveluihin liittyvät toiminnot. Oppilaitoksen on virtuaalista oppimisalustaa valitessaan valittava joko kaupallisen toimijan tarjoama kokonaisratkaisu tai kehittäjäyhteisön ylläpitämä ilmaisratkaisu, jonka kehittämiseen myös oppilaitoksen itsensä on mahdollista osallistua. (Keränen & Penttinen 2007, 28 – 29.)

Oppimisympäristön tarkoituksena on edistää opiskelijan oppimisprosessin edellytyksiä. Tämä saavutetaan huomioimalla opiskelijoiden tarpeet ja mielenkiinnon kohteet sekä tekemällä oppimisesta tavoitteellista. Oppimista edistäviä tekijöitä ovat muiden muassa vuorovaikutusmahdollisuudet, ohjaus ja tuki. (Vahtivuori 2001, 82.)

Verkko-opettamisen keskittäminen auttaa luodun materiaalin yhtenäistämisessä. Yksi käytettävä väline poistaa myös tarpeen käyttää useita eri kommunikaatiovälineitä ja sijainteja materiaalia jaettaessa. Mahdollisuus aineistojen keskittämiseen korostuu modulaarisissa oppimisalustoissa, joissa yhtenä tiedostona sijaitseva aineisto voidaan liittää useaan eri kurssipohjaan. Näin aineistoa muutettaessa päivitys on tarpeen kohdistaa vain itse tiedostoon kurssipohjien sijaan. (Kalliala 2002, 109 – 112.) Oppimisalustoiden käyttömahdollisuudet ovat monipuolisia. Oppimisalustaa käyttämällä voidaan oppimateriaalin jaon lisäksi muun muassa järjestää kurssin tiedotus sekä arviointien ja palautteiden anto. (Keränen & Penttinen 2007, 20.)

2.3 Oppimismateriaalit

Oppimisprosessissa opettajan tavoitteena on löytää keino tiedon ja viestien välittämiseen opiskelijalle niin, että viestiin sisällytettävä tieto välittyy opiskelijalle omaksuttavassa muodossa. Erilaisten teknologioiden käyttö voidaan ottaa osaksi opetusta helpottavana, havainnollistavana tai muuten oppimisprosessia edistävänä

tekijänä. Tekniikoiden kehittyminen on parantanut edellytyksiä luoda ja tarjota entistä monipuolisempaa oppimismateriaalia.

Oppimismateriaalit voidaan jaotella niiden käyttämien tekniikoiden perusteella. Tällöin voidaan muodostaa kolme erillistä oppimateriaalityyppiä, jotka ovat: selainpohjaiset materiaalit, erilliset tiedostot sekä itsenäiset ohjelmat. Toinen oppimateriaaleista käytettävä nimitys on oppimisaihiot. Oppimisaihioita kehitettäessä tavoitteena on tuottaa materiaalia, joka on vapaasti yhdisteltävissä muuhun olemassa olevaan aineistoon. (Keränen & Penttinen 2007, 5 – 6.)

Verkko-oppimismateriaaleina voidaan käyttää esimerkiksi tekstiä, ääntä, kuvaa ja videota. Oppimismateriaaliksi käsitetään myös opiskelijoiden yksin tai yhdessä opettajan kanssa oppimisprosessin aikana tuottamat esitelmät, projektityöt ja keskustelut. (Kalliala 2002, 14.) Verkko-opetukseen käytettävällä verkkomultimedialla tarkoitetaan yleisimmin WWW-sivuihin yhdistettyä ääntä, videota ja animaatiota. Multimedian tarkastelu edellyttää joko käyttötarkoitukseen sopivan selainlaajennuksen asennusta tai selaimesta erillistä ohjelmaa. (Keränen & Penttinen 2007, 9.) Verkkomateriaalia tuotettaessa on tärkeää varmistua siitä, että tuotettava materiaali on todella niiden henkilöiden käytettävissä, joille materiaali on suunnattu. Mikäli käytettävä laitteisto ohjelmistoinen ei tue kyseistä materiaalia, vaaditaan lisäasennusten tekemistä, mikä saattaa aiheuttaa ongelmia osaamistason tai käyttöoikeuksien puutteen vuoksi. (Flinkman ym. 2006, 82.)

Visualisoinnilla parannetaan oppimisympäristön ilmaisuvoimaa tarjoamalla vaihtoehtoisia ratkaisuja erilaisten käyttäjäryhmien tarpeisiin. Visuaalista havainnollistamista ei tule kuitenkaan väärinkäyttää niin, että huomio kiinnittyy epäolennaisuuksiin. (Meisalo ym. 2003. 244.) Liikkuvan kuvan käyttö verkko-opetuksessa edellyttää opettajilta ja opiskelijoilta kuvallisten mediataitojen kehittyneisyyttä. Visuaalisilla elementeillä saavutetaan kokonaisvaltaisempi oppimisprosessi sekä tehostunut vuorovaikutus. (Vahtivuori 2001, 81.)

Interaktiivisuudella on oma osuutensa motivoivan oppimisprosessin muodostamisessa. Interaktiivisuuden kautta saavutettava sosiaalisuus voidaan toteuttaa verkko-

opiskelussa esimerkiksi video- tai audiotekniikoita hyödyntämällä. (Yli-Luoma & Pirkkalainen 2005, 11 – 12.) Sen lisäksi että videota voidaan käyttää verkko-opetuksessa itsenäisenä elementtinä, voidaan se myös yhdistää muiden medioiden käyttöön tai kommunikaatioon. Videolla pyritään usein luomaan konteksti oppimisprosessiin tai motivoimaan opiskelijaa. (Silander & Koli 2003, 76.)

Videoaineistoja käytettäessä on huomioitava käyttöön kohdistuvat rajoitukset. Jatkuvana videokuvana lähetettävä pitkäkestoinen kuvamateriaali vaatii käyttötarkoitukseen sopivan palvelimen. Usein kurssimateriaalit ovatkin lyhyehköjä ja niiden tarkoituksena on vaihtoehtoisen esitystavan tarjoaminen. (Kalliala 2002, 69 – 70.) Verkon välityksellä seurattavan yhdestä kohteesta välitettävän kuvan ja äänen lisäksi opetuksessa voidaan hyödyntää videoneuvottelua, joka mahdollistaa reaaliajassa tapahtuvan kommunikaation kahden tai useamman henkilön kesken.

3 MOODLE VIRTUAALISENA OPPIMISYMPÄRISTÖNÄ

Maailman käytetyin verkko-opetusympäristö kehitettiin alunperin tarjomaan oppilaitoksille aiempia vastaavia sovelluksia parempi työväline verkko-opetuksen järjestämiseen. Helppous kurssien ja niiden sisällön toteuksen osalta oli keskeinen tavoite lähdettäessä kehittämään Moodlea. (Karevaara 2009, 15.) Moodle on *open source* –sovelluksena oppilaitoksille kustannustehokas ja helposti käyttöön otettavissa vaihtoehto. Oppilaitoksissa on mahdollista päättää itsenäisesti versioiden päivityksistä sekä sovellukseen mahdollisesti tehtävistä korjauksista ja parannuksista. (Flinkman ym. 2006, 73.)

3.1 Käytettävät materiaalit

Moodle tarjoaa perinteisistä verkkokursseista poikkeavan oppimiskokemuksen siinä, että se mahdollistaa vapaassa järjestyksessä ja lukuisin eri tavoin hyödynnettävän materiaalivalikoiman. Interaktiivisuus kyetään Moodlessa saavuttamaan esimerkiksi kyselyjen, keskustelujen ja tiedostojen jaon avulla. (Rice IV 2006, 5.) Moodle on suunniteltu toteuttamaan interaktiivisen oppimisen periaatteita. Oppimisfilosofian mukaan paras oppimistulos saavutetaan interaktiivisella materiaalilla, jota opiskelijat

tuottavat ja jakavat toisilleen. Perinteisiä oppimismateriaaleja Moodlessa ovat esimerkiksi staattiset teksti- ja internetsivut sekä linkit muualla internetissä sijaitseviin kohteisiin. Interaktiiviset materiaalit tarjoavat puolestaan opiskelijoille mahdollisuuden osallistua opiskeluun materiaalia muokkaamalla. Näitä materiaaleja ovat muiden muassa palautettavat tehtävät, kyselyt ja testit. Kolmannen eli sosiaalisen materiaalityypin muodostavat chatit, keskustelalueet, sanastot, wikit ja ryhmätyöt. Koska Moodle on suunniteltu juuri vapaassa järjestyksessä läpikäytävää materiaalia käyttäväksi oppimisvälineeksi, ei se välineenä tarjoa juurikaan mahdollisuuksia lineaarisen kurssikokonaisuuden luomiseen. (Rice IV 2006, 9 – 10.)

Kurssien pääasiallinen sisältö Moodlessa koostuu kursseihin liitettävistä aineistoista ja aktiviteeteista. Moodlen oletuksena sisältyvät aineistotyypit on listattu taulukossa 1. Aineistot ovat staattisia ja läpikäytäviä materiaaliaineistoja. Aktiviteetit puolestaan on suunniteltu ensisijaisesti aktiiviseen ja vuorovaikutteiseen opiskeluun. (Karevaara 2009, 46 – 48.)

TAULUKKO 1. Aineistotyypit Moodlessa

Aineistot
Tekstisivu
Web-sivu
Linkki
Hakemisto
IMS-sisältöpaketti
Otsikko

Moodlen vakioasennuspaketissa tulevat aineistotyypit kattavat keskeisimmät staattisen materiaalin esitystavat. Aineistojen sisältämä materiaali on aktiviteetteihin verrattuna rajatumpaa ja suppeampaa. Mahdollisten lisäosien voidaankin näin luontevasti ajatella kuuluvan ennemmin aktiviteetteihin kuin aineistoihin. Moodlen perusaktiviteettityypit on esitetty taulukossa 2. Aktiviteetteja tarkasteltaessa voidaan todeta, että kyseiset materiaalityypit edellyttävät yleistettynä opiskelijalta aktiivista osallistumista mahdollisesti opettajan rajaamana ajankohtana.

TAULUKKO 2. Aktiviteettityypit Moodlessa

Aktiviteetit
Chat
Keskustelualue
Kysely
Oppitunti
Sanasto
Scorm
Tehtävä
Tentti
Tietokanta
Valinta
Wiki

Kolmannen elementtityypin Moodlessa muodostavat lohkot. Lohkojen merkitys kurssinäkymässä on kuitenkin lähinnä kurssin tapahtumien ja edistymisen seuraamista helpottava, vaikka ne voivatkin sisältää myös samankaltaista materiaalia kuin aineistot ja aktiviteetit. Varsinaisen opiskeluaineiston sijaan lohkot sisältävät kuitenkin lähinnä ylläpitoon ja tiedotukseen liittyvää sisältöä.

3.2 Tekninen rakenne

Moodle voi toimia millä tahansa PHP-kieltä ja tietokantaa tukevalla web-palvelimella. Suositelluin vaihtoehto on kuitenkin Apache/MySQL-kokoonpano, jonka toimivuus ja tuki ovat parhaimmalla tasolla. Moodlen tiedot ja tiedostot tallennetaan kolmeen eri kohteeseen. Sovelluksen omat tiedostot sijaitsevat omassa kansiossaan web-palvelimella. Käyttäjien sivustolle lataamat tiedostot puolestaan omassa sijainnissaan datakansiossa. Tämän lisäksi tiedot esimerkiksi käyttäjistä ja kursseista tallennetaan Moodlen tietokantatauluihin. (Rice IV 2006, 27.)

Moodlen asennuskansio sisältää tuhansia tiedostoja sekä satoja kansioita, joten on hyvä olla selvillä ainakin hakemistorakenteen perusrakenteesta. Esimerkiksi lisäosia asennettaessa ja kehitettäessä ei useidenkaan hakemistojen sisältöihin ole tarvetta sen tarkemmin tutustua, mutta jo esimerkiksi otettaessa käyttöön kustomoituja teemoja, on

näiden asennussijainnit tiedostettava. Taulukossa 3 on esitetty keskeisimmät Moodlen asennuskansion hakemistoista sekä selitykset niiden käyttötarkoituksista ja sisällöistä.

TAULUKKO 3. Moodlen hakemistorakenne (Keränen & Penttinen 2007, 265.)

Hakemisto	Tehtävä
admin	Ylläpitomoduuli
auth	Käyttäjien tunnistusmoduulit
blocks	Laajennukset
calendar	Kalenterimoduuli
course	Kurssinhallintamoduulit
blog	Blogi-moduulin toiminnot
doc	Dokumentaatio
files	Tiedostojenhallinta moduuli
lang	Kielipaketit
lib	Ydinkirjastot
login	Kirjautumiskoodi
mod	Kurssimoduulit
pix	Käyttöliittymän kuvakkeet
theme	Ulkoasuteemat
user	Käyttäjienhallinta

Valtaosa Moodlen perusjakelupaketin asennuksen tietokantaan ajamista noin 200:sta taulusta on aktiviteettien eli moduulien käyttämiä. Moduulien taulut ryhmitetään niin, että taulujen nimiin liitetään etuliite. Muut keskeisimmästä tauluista voidaan jakaa sisältämiensä tietojen mukaan seuraaviin kategorioihin:

- Aktiviteetit
- Käyttäjät
- Roolit
- Kurssit
- Ryhmät

Edellämainittujen kategorioiden lisäksi tietokanta sisältää lähinnä ylläpitoon liittyviä tauluja, joista keskeisimpiä ovat loki-, varmuuskopiointi- sekä tilastointitaulut.

Ohjelmakoodiltaan Moodle perustuu oppimisalustan kehittäjien luomiin oliokirjastoihin. Oliokirjastot on jaettu käyttötarkoitustensa mukaisesti erillisiin *.lib*-päätteisiin tiedostoihin. Hyödyntämällä olemassa olevia funktioita säilytetään yhteneväisyys esitystavassa ja yhteensopivuus esimerkiksi erilaisten tietokantaratkaisujen kanssa sekä tehostetaan kehitysprosessia. Lisäosien kehityksen kannalta tärkeimmät oliokirjastot on esitetty taulukossa 4.

TAULUKKO 4. Moodlen keskeisimmät oliokirjastot

datalib.php	Funktiot tiedon keräämiseen tietokannasta
dml.lib.php	Tietokantakyselyjen ajamisfunktiot
moodlelib.php	Moodlen perusfunktiot
weblib.php	Moodlen ulkoiseen olemukseen liittyvät funktiot

Funktioiden käyttöä havainnollistaa esimerkki parametrien käsittelystä, jolle on monien muiden toimintojen tavoin Moodleen kehitetty oma funktionsa. Ennen jokaista funktiota on koodiin liitetty kommenttina funktion käyttötarkoitus sekä toimintaperiaate:

- * *Returns a particular value for the named variable, taken from*
- * *POST or GET, otherwise returning a given default.*
- * *This function should be used to initialise all optional values*
- * *in a script that are based on parameters. Usually it will be*
- * *used like this:*
- * *\$name = optional_param('name', 'Fred');*
- * *@param string \$parname the name of the page parameter we want*
- * *@param mixed \$default the default value to return if nothing is found*
- * *@param int \$type expected type of parameter*
- * *@return mixed*

Tarvetta funktion varsinaisen koodin tarkasteluun ei käytännössä ole, ellei ole tarvetta muokata jo olemassa olevia funktioita omiin käyttötarkoituksiin. Kommenteista selviävät muun muassa vaihtoehtoiset funktiot. Tärkeimpänä osuutena voidaan pitää selvitystä funktion saamista parametreista. Ohjelmakoodissa käytetyn funktion paikallistamista oliokirjastoista helpottaa sivusto *xref.moodle.org*, jolla olevalla hakutoiminnolla on mahdollista etsiä sijainti, jossa funktion toiminta esitellään. Varsinainen funktio on esitetty oliokirjaston koodissa seuraavasti:

```
function optional_param($parname, $default=NULL, $type=PARAM_CLEAN)
```

Seuraavassa esimerkissä esitetään funktion käyttö niin, että valinnainen muuttuja *\$name* saa merkkijonomuotoisen arvon joko *POST*- tai *GET*-metodilla välitetystä *name*-parametrasta:

```
$name = optional_param('name', '', PARAM_ALPHA);
```

Funktion parametreiksi annetaan *\$_POST*- tai *\$_GET*-taulukon muuttujan nimi, oletusarvo - joka esimerkissä on jätetty tyhjäksi - sekä muuttujan tyyppi.

Moodlen kustomointimahdollisuudet perustuvat koodin selkeyteen sekä teknisten toteutusten dokumentaatioon. Modulaarisuus ja avoin lähdekoodi rohkaisevat oppimisalustan käyttäjiä kehittämään uusia liitännäisiä sekä jakamaan näitä muiden käyttäjien kanssa. Moodlen dokumentaariosivustolla esitetty materiaali on suurelta osin keskeneräistä, mutta apuvälineitä kehitystyön käynnistämiseen on kuitenkin melko kiitettävästi tarjolla.

Moodlen vapaiden kustomointimahdollisuuksien myötä joudutaan huomioimaan versiopäivitysten vaikutukset oppilaitoksen omiin tarkoituksiinsa kehittämiin ominaisuuksiin. Ilmaisen oppimisympäristön kääntöpuoli on, että tarjolla ei ole palvelua uuden version parannettujen ominaisuuksien ja edellistä versiota varten kehitettyjen ominaisuuksien yhdistämiseen. (Heynderickx 2006.) Moodlen mukana tulevien ominaisuuksien lisäksi moduuleja on ladattavissa Moodlen virallisilta sivuilta, joille kehittäjät lisäävät kehittämiään ominaisuuksia. (Rice IV 2006, 215.)

Lisättävien moduulien ongelmia kuitenkin monissa tapauksissa ovat, ettei yhteisön tuottamien lisäosien käytettävyydestä ole täyttä takuuta sekä monikielisyyden puuttuminen. (Keränen & Penttinen 2007, 268.)

Moduulien asentaminen on Moodlessa tehty hyvin yksinkertaiseksi. Sivuston hallinnoijan tulee muokkaustilassa ollessaan ainoastaan siirtää lisättävä moduulikansio Moodlen omaan moduulikansioon. Tämän jälkeen hallinnoijan siirtyessä hallinnoinnin etusivulle Moodle löytää automaattisesti lisätyn moduulin ja päivittää moduulin edellyttämät tiedot tietokantaansa. Moduulin asennus poistetaan puolestaan valitsemalla poisto hallinnointityökaluista ja poistamalla tämän jälkeen aiemmin lisätty moduulikansio. (Rice IV 2006, 217 – 218.)

4 FLASH-FORMAATTI WEB-VIDEOIDEN JULKAISUSSA

Flash-ohjelmat jakautuvat kahteen eri tyyppiin. Soittimia käytetään Flash-videoita toistossa, ja tuotanto-ohjelmilla, joista yksi esimerkki on Adoben Flash Professional CS3, tehdään Flash-leikkeitä. Loppukäyttäjän näkemä Flash-soitin videoleikettä katsoessaan on siis vain toinen puoli Flash-videoiden käytöstä. Flash-soitin on joko oma itsenäinen ohjelmansa tai selaimessa toistettava ja HTML-koodiin upotettu elementti. Flashin asema multimediaformaattina on nykyisin hyvin vahva. Internetiä käyttävistä tietokoneista 90 % sisältää Flash-leikkeiden katselussa käytettävän ohjelman. Flashin yksi vahvuus on, että se tukee erilaisia videotyyppejä. Sekä tavallisten videoleikkeiden että interaktiivisten multimedialeikkeiden teko on toteutettavissa Flash-formaatissa. Nämä erilaiset mahdollisuudet ovat johtaneet siihen, että Flashia käytetään nykyisin moneen eri tarkoitukseen. Näitä ovat esimerkiksi laajasti nähtävissä oleva internetmainostus sekä internetsivujen käyttöliittymien toteutus. Flashia käytetään myös Microsoftin Power Point -ohjelman tapaan esitysten valmistamiseen. Viime aikoina Flashin suosio videoformaattina on kasvanut entisestään Web 2.0 -ilmiön myötä. Monet sosiaaliset internetpalvelut kuten YouTube ja MySpace hyödyntävät Flashia videon ja äänen välittämisessä käyttäjille. (Paananen 2008, 6 – 7.)

4.1 Flash-formaatin toisto

Sosiaalisten medioiden lisäksi Flash Videota hyödynnetään internetin välityksellä välitettävien TV-ohjelmien jakelussa. Yksi Flash Videon tärkeimmistä eduista on, että se on mahdollista pakata pienempään kokoon, jolloin video on jaettavissa verkon yli nopeammin eikä sen säilytys vaadi liikaa fyysistä levytilaa. Pakkaustoimenpiteestä käytetään nimitystä *encoding* ja ennen toistoa suoritettavasta purusta nimitystä *decoding*. Videotiedoston koon pienentämiseen on myös muita mahdollisuuksia. Videokuvan fyysistä kokoa eli resoluutiota voidaan pienentää. Vaihtoehtoisesti voidaan vähentää videokuvan ruutujen määrää normimäärästä, joka on 25 ruutua sekunnissa. Flash Video (*.flv*) on edullisin tiedostomuoto tuotettaessa Flash-videoita kokoa pienentävän pakkauksen ansiosta. Flash Videon saaminen esimerkiksi videokameralla kuvatusta kuvamateriaalista edellyttää tiedostomuodon konvertointia, koska useimmat laitteet eivät tallenna videokuvaa kyseisen tiedostomuotoon. (Paananen 2008, 102 – 103.)

Palveluntarjoajalla on kolme mahdollisuutta toteuttaa videoiden latauspalvelu. Perinteisessä tavassa videotiedosto ladataan käyttäjän päätteelle progressiivisesti. *Streaming*-vaihtoehdossa mediapalvelin puolestaan vastaa käskyyn toistaa tiedosto tietystä kohdasta alkaen. Tässä vaihtoehdossa tiedosta ei ladata käyttäjän koneelle, vaan tiedosto toistetaan käyttäen RTMP-protokollaa. Parhaiten tunnettuja mediapalvelimia ovat Adoben Flash Media Server, Wowza Media Server sekä avoimeen lähdekoodiin pohjautuva Red5. (Knowles 2007.)

Perinteisellä tavalla progressiivisesti ladattava tiedosto välittyy käyttäjän päätteelle HTTP-protokollan välityksellä. Virtausjaeltavan videon välitysprotokolla vaihtelee välitettävän videotyyppin mukaan. Esimerkiksi Windows Median käyttämä protokolla on RTMP:n sijaan joko MMS tai RTSP. (Colburn 2010a.)

Virtausjakelu edellyttää erillistä mediapalvelinta, jonka toimintoja ei sellaisinaan ole sisällytetty oppimisalustoihin. Virtana jaettavaa tiedostoa ei voida tallentaa tai kopioida. Virtausjakelun yleinen käyttötarkoitus on reaaliaikaisen videokuvan välitys verkon kautta. Videotiedostoja jaettaessa on huomioitava yhteysnopeuksien

määrittämät latausajat. Esimerkiksi 10 Mt:n kokoisen tiedoston latautuminen kestää yleisesti käytettävällä yhden Mt:n yhteydellä minuutin ja 18 sekuntia. (Keränen & Penttinen 2007, 191 – 192.)

Uudempi jakelumenetelmä on niin kutsuttu *pseudo streaming*, jonka käyttöön perustuu muiden muassa YouTube-sivuston toiminta. Kyseisen menetelmän suurin etu on, että lataus tapahtuu kuten perinteisessä progressiivisessa mallissa, mutta sillä erolla, että toisto voidaan aloittaa jo ennen latauksen valmistumista kokonaan. Kyseisessä tavassa Flash-soittimesta välitetään tieto siitä hetkestä, josta alkaen videota halutaan toistaa, minkä jälkeen soittimeen ladataan puskuroinnin edellyttämä osa videosta. Menetelmä tarjoaa progressiivista tapaa käyttäjäystävällisemmän menetelmän ilman erillisen mediapalvelimen tarvetta. (Armitage 2009.)

4.2 Flash-videoiden liittäminen web-sovelluksiin

Toistaiseksi Flash-videoiden liittäminen WWW-sivulle edellyttää erillisen Flash-liitännäisen olemassaoloa. Sen sijaan tulevaisuudessa nykyinen liitännäistä edellyttävä menetelmä on tarkoitus korvata uuteen HTML5-syntaksiin määriteltävällä videoelementillä. (Coldewey 2009.) Kunnes uusi HTML-spesifikaatio on otettavissa käyttöön, toteutetaan Flash-soittimen liittäminen web-sivulle kahden JavaScript-skriptin avulla. Näistä ensimmäinen *swfobject.js* etsii, löytyykö käyttäjän koneelta Flash-asennusta. Toiseen skriptiin määritellään puolestaan palvelimelta löytyvän soittimen sijainti sekä halutut ominaisuudet eli *FlashVars*-parametrit soittimelle. Määriteltäviä ominaisuuksia ovat muiden muassa:

- *file* – toistettava tiedosto
- *image* – soittimessa näytettävä aloituskuva
- *streamer* – virtausjakelun lähde

Näiden lisäksi voidaan määritellä erinäisiä ulkoasullisia ja toiminnallisia ominaisuuksia. JavaScript-skriptissä soittimen ominaisuudet määritellään niin, että

luodaan uusi soitinolio, jolle määritellään halutut parametrit kuten soittimen koko. Tämän jälkeen olion muut ominaisuudet määritellään *addParam*-metodilla:

```
var s1 = new SWFObject("player.swf","mpl","720","406","9");  
s1.addParam("flashvars","&file=stream.flv&image=moodle.png");
```

Varsinainen Flash-soitin on web-sivulle upotettava SWF-tiedosto. Vaihtoehtoisia soittimia on olemassa useita, tunnetuimpia näistä ovat JW Player sekä Flow Player. Soitintiedosto sisältää soittimen käytettävissä olevat toiminnot ja on ulkoasultaan muokattavissa. (Colburn 2010b.)

Web-kamerakuvan tallentaminen palvelimelle edellyttää, että käyttäjä sallii Flash-soittimen ottavan yhteyden käyttäjän web-kameraan. Kuvassa 1 näkyvässä ikkunassa Flash-liitännäinen ilmoittaa käyttäjälle sivuston (*localhost*) haluavan ottaa yhteyden web-kameraan. Käyttäjälle on täten palvelun yhteydessä tehtävä selväksi, mitä vahvistuksen hyväksyminen tarkoittaa.

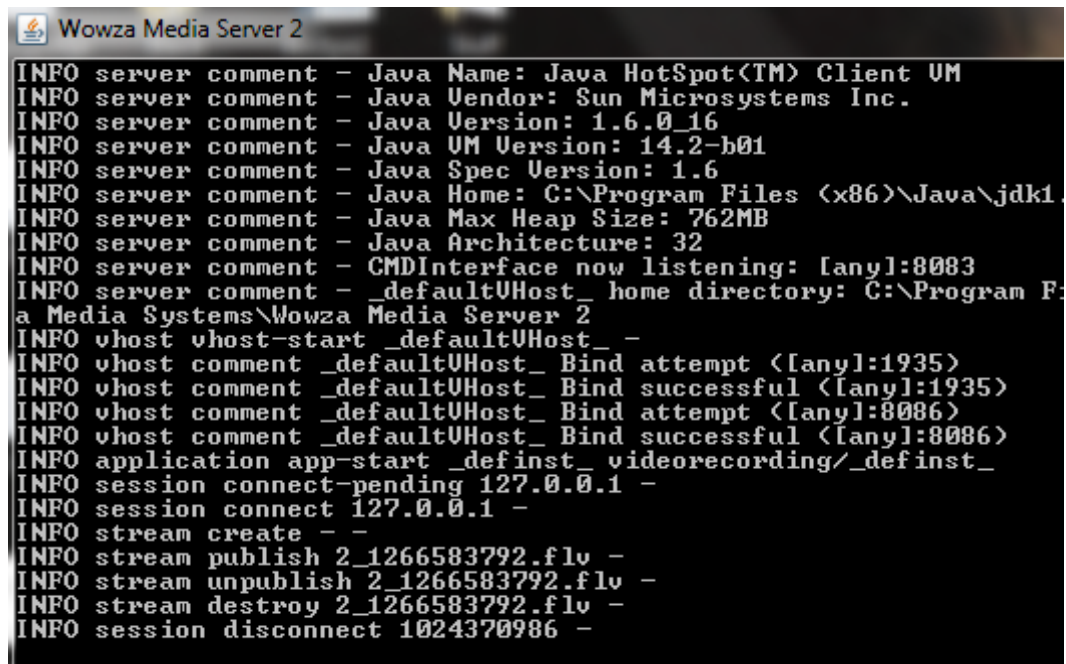


KUVA 1. Yhteyden hyväksyntä

Jotta soittimen liittäväan skriptiin voidaan määritellä tallentavan palvelimen sijainti, on palvelimen sovelluskansioon luotava uusi kansio. Kansiota käytetään yhteyden ottamiseen web-sovelluksesta niin, että IP-osoitteen jälkeen ilmoitetaan sovelluskansioon luotu uusi kansio:

rtmp://127.0.0.1/videorecording

Kun yhteys palvelimeen on hyväksytty ja muodostettu, tallennettava video tallentuu palvelimen asennuskansiossa olevaan *content*-kansioon.



```

Wowza Media Server 2
INFO server comment - Java Name: Java HotSpot(TM) Client VM
INFO server comment - Java Vendor: Sun Microsystems Inc.
INFO server comment - Java Version: 1.6.0_16
INFO server comment - Java VM Version: 14.2-b01
INFO server comment - Java Spec Version: 1.6
INFO server comment - Java Home: C:\Program Files (x86)\Java\jdk1.6.0_16
INFO server comment - Java Max Heap Size: 762MB
INFO server comment - Java Architecture: 32
INFO server comment - CMDInterface now listening: [any]:8083
INFO server comment - _defaultVHost_ home directory: C:\Program Files (x86)\Wowza Media Systems\Wowza Media Server 2
INFO vhost vhost-start _defaultVHost_ -
INFO vhost comment _defaultVHost_ Bind attempt ([any]:1935)
INFO vhost comment _defaultVHost_ Bind successful ([any]:1935)
INFO vhost comment _defaultVHost_ Bind attempt ([any]:8086)
INFO vhost comment _defaultVHost_ Bind successful ([any]:8086)
INFO application app-start _definst_ videorecording/_definst_
INFO session connect-pending 127.0.0.1 -
INFO session connect 127.0.0.1 -
INFO stream create -
INFO stream publish 2_1266583792.flv -
INFO stream unpublish 2_1266583792.flv -
INFO stream destroy 2_1266583792.flv -
INFO session disconnect 1024370986 -
  
```

KUVA 2. Palvelimen toiminta

Kuvassa 2 on esitetty mediapalvelimen toiminta Flash-sovelluksen ottaessa yhteyden palvelimeen. Komentoikkunassa näytetään saapuva yhteys, yhteyden hyväksyntä käyttäjän päätteellä sekä vastaanotettavan virtausjakelun julkaisu. Tallennuksessa käytettävä tiedostonimi on määritelty Flash-soittimen skriptissä.

5 VIDEOAINEISTOMODUULIN KEHITTÄMINEN MOODLEEN

Moodlesta vakio-ominaisuuksina löytyvät opetusmateriaalit sisältävät mahdollisuuden informaation esittämiseen lähinnä tekstinä ja kuvina. Äänen ja videokuvan lisääminen kursseille on ollut mahdollista lähinnä luomalla linkki ulkopuolisilla sivustoilla sijaitseviin palveluihin. Moodle ei ole itsessään tukenut opettajan tai opiskelijoiden itsensä kuvaaman videoaineiston hyödyntämistä opetuksessa. Mikäli käyttäjän tietoteknisissä taidoissa on huomattavasti kehittämisen varaa, voidaan useiden eri

sisällöntuottopalveluiden hallinta kokea kohtuuttoman paljon aikaa ja vaivaa vaativaksi.

5.1 Toimintaperiaate

Lähtökohta järjestelmän toteuttamiseen oli pyrkimys luoda järjestelmä, joka tallentaa opettajan web-kameran kuvaaman videon Moodlen yhteydessä olevalle palvelimelle reaaliajassa. Käyttöliittymä oli tarpeen säilyttää mahdollisimman yksinkertaisena, jotta kynnys ominaisuuden käyttämiseen säilyisi riittävän matalana. Tämä tarkoitti sitä, että videoiden teknisten ominaisuuksien määrittely ei saisi edellyttää videon lisääjältä minkäänlaisia toimia, vaan videot tallennetaan samassa ennalta määritellyssä formaatissa. Moduuli kehitettiin Mikkelin ammattikorkeakoulun käyttöönottamaan Moodlen 1.9-versioon.

Keskeinen haaste moduulin toteutuksessa oli Flash-soittimen upottaminen Moodlen kurssinäkömään sekä toistettavan videon id-tunnuksen välittäminen Flash-soittimelle. Ongelman ratkaisu on esitetty seuraavassa:

```
$objectscript = file_get_contents("object.php");
$object = str_replace("file=stream.flv", "file=".$i$filename.".flv", $objectscript);
```

Koodissa mainittava *object.php*-tiedosto sisältää luvussa 4.2 mainitut JavaScript-skriptit. PHP:n *file_get_contents*-funktiolla tiedoston sisältö luetaan *string*-tyypin muuttujaan. Funktiolla *str_replace* puolestaan välitetään merkkijonoon toistettavan videon tiedostonimi. Tämän jälkeen muutettu muuttuja voidaan esittää web-sivulla. Samaa menetelmää käytettiin myös tallentavan Flash-soittimen liittämässä moduuliin. Tällöin ainoastaan soittimen *FlashVars*-parametrien arvot ovat toiset.

Vaihtoehtoisesti parametrien välitys linkistä soittimeen olisi voitu toteuttaa JavaScript-skriptillä. Pelkän yksittäisen tiedostonimen lisäksi soittimelle on mahdollista välittää myös XML-muodossa oleva soittolista useammasta toistettavasta videosta. Moduulin tässä kehitysvaiheessa ei kuitenkaan ollut tarpeellista välittää tieto kuin yhdestä videosta kerrallaan.

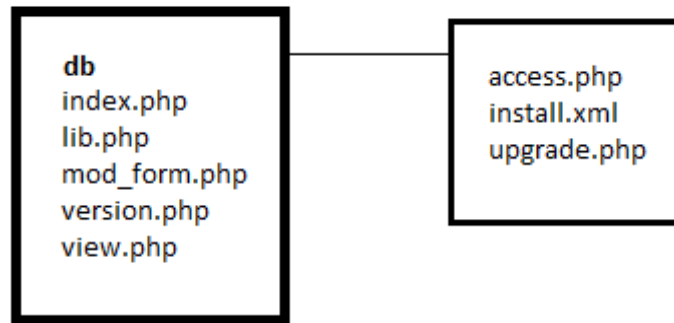
Lähdettäessä kehittämään aineistomoduulia Moodle-ympäristöön valittavana oli kaksi erilaista lähestymistapaa. Aineistot jakautuvat rakenteellisesti kahteen tyyppiin: aineistoihin ja aktiviteetteihin. Ensimmäiseksi oli valittava, kumpaa kehitettävästä aineistosta lähdetään tekemään. Resurssin tyyppi määritellään skriptissä *moodle/mod/resource/view.php*:

```
require ($CFG->dirroot.'/mod/resource/type/'.$resource->type.'/resource.class.php');
$resourceclass = 'resource_'.$resource->type;
$resourceinstance = new $resourceclass($cm->id);
```

Ajatus lisäosan toteuttamisesta aineistona kuitenkin hylättiin. Tehtävän kannalta mielekkäämpi vaihtoehto oli videojärjestelmän toteuttaminen moduulina, koska moduulien käyttöönotto on teknisesti aineistojen käyttöönottoa yksinkertaisempi toteuttaa. Moduulit ovat myös vapaammin modifioitavissa ja sopivat laajuutensa vuoksi paremmin videoiden vaatiman käyttöliittymän sisällyttämiseen etenkin siinä tapauksessa, että moduulin ominaisuuksia halutaan tulevaisuudessa laajentaa.

Moodleen on määriteltynä tapa käyttäjien jaotteluun näille sallittujen käyttöoikeuksien mukaan. Uutta moduulia toteutettaessa oli keskeistä määritellä moduuliin halutut toiminnot käyttäjäluokkoittain. Moduulin toiminnot oli suunnitteluvaiheessa määritelty niin, että videoiden lisäysmahdollisuus haluttiin sallia ainoastaan kursseja pitävälle opettajille. Mahdollisuus lisäämisen sallimiseen myös opiskelijoille olisi ollut olemassa, mutta muun muassa tallennuskapasiteetin aiheuttamat rajoitukset siirsivät tämän vaihtoehdon toteuttamisen myöhemmän harkinnan varaiseksi.

Moduulien kehittäjille Moodle tarjoaa sivuillaan esimerkin siitä, millaisia moduulit rakenteeltaan ovat. Kuvassa 3 on esitetty Moodle-moduulin perusrunko, joka sisältää moduulin toiminnan kannalta välttämättömät PHP-skriptit sekä tietokannan määrittelevän XML-dokumentin.



KUVA 3. Moduulin perusrunko

Videoaineistomoduulin kehittäminen ei edellyttänyt suuria muutoksia yleisen moduulirakenteen perusrunkoon. Flash-objektin liitântään ja käyttöön liittyvät skriptit sijoitettiin kuitenkin omiin tiedostoihinsa. Muilta osin luodun moduulin rakenne seuraa Moodle-moduulien yleisiä periaatteita:

- *index.php* listaa kaikki moduulin instanssit kyseisellä kurssilla. Moduulia kehitettäessä kyseiseen skriptiin ei tarvittu muutoksia.
- *lib.php* sisältää moduulin käyttämät funktiot, jotka jakautuvat skriptissä kahteen eri tyyppiin. Oletuksena skriptistä löytyvät funktiot liittyvät moduulin yleisiin ylläpidollisiin toimiin kuten lisäyksiin, poistoihin ja päivityksiin. Kehittäjän luomat muut moduulin tarvitsemat funktiot sisällytetään skriptiin, mutta ne voidaan jakaa tarvittaessa selkeyden vuoksi jakaa myös omiin kirjastoihinsa.
- *mod_form.php* on instanssia lisättäessä ja muokattaessa näytettävä sivu, joka sisältää moduulien määrittelyvaihtoehdot, joista valikoimalla lisääjä tai muokkaaja muuttaa moduulin sisältöä haluamakseen.
- *version.php* sisältää moduulin versionumeron, jota Moodlen ajastetut toiminnot hyödyntävät.
- *view.php* on moduulin pääasiallisen näkymän esittävä skripti. Skriptissä kutsutaan PHP:n ehto- ja silmukkalauseita hyödyntäen Moodlen ja moduulin oliokirjastoissa määriteltyjä funktioita toiminnallisuuden toteuttamiseksi.

Kansiossa *db* olevaan *access.php* määritellään moduulin toiminnassaan käyttämät jaottelut eri käyttäjäryhmien kesken. Seuraavassa esimerkissä on luotu käyttöoikeustapaus nimeltä *allowvideos*:

```
$mod_webcam_capabilities = array(
    'mod/webcam:allowvideos' => array(
        'captype' => 'read',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'teacher' => CAP_ALLOW));
```

Määrittelyn tuloksena saadaan eriteltyä sivulle kussakin tapauksessa tulostettavat tiedot käyttäjäluokan mukaan. Esimerkissä luodun oikeusluokan tarkoituksena on sallia videoiden lisääminen ainoastaan *teacher*-käyttäjäluokalle. Oikeuden tarkistus moduulin *view*-skriptissä on esitetty seuraavassa:

```
has_capability('mod/guidance:allowedediting', $context)
```

Käytetty *has_capability*-funktio on määritelty Moodlen kirjastohakemistossa sijaitsevassa *accesslib.php*-tiedostossa. Muuttuja *\$context* on tätä ennen saatu samassa kirjastossa määritellyllä *get_context_instance*-funktioilla:

```
$context = get_context_instance(CONTEXT_MODULE, $cm->id);
```

5.2 Tietokantarakenne

Moodlen oma työkalu XML-dokumenttien muokkaamiseen on sivun hallinnoitsijalle näkyvissä olevasta ylläpitovalikosta löytyvä *XMLDB editor*. Editointityökalu tarjoaa mahdollisuuden muokata tietokantadokumentteja graafista käyttöliittymää käyttäen XML-kielen manuaalisen muokkaamisen sijaan. Moduulin perusrungon mukaisesti tietokantamäärittelyt löytyvät *install.xml*-dokumentista, joka sijaitsee moduulin omassa kansiossa sijaitsevasta *db*-kansioista. Tietokantataulut asentava XML-

dokumentti on jaettavissa kahteen perusmäärittelytyyppiin: *tables*- ja *statement*-elementteihin. Luotavan taulun attribuutit määritellään seuraavasti:

```
<TABLE NAME="webcam" COMMENT="Main table" NEXT="webcam_video">
```

Tables-elementin sisällä olevassa *fields*-elementissä luodaan samalla tavoin moduulin tietokantarakenne määrittelemällä kustakin taulusta löytyvät sarakkeet ja näiden ominaisuudet:

```
<FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" SEQUENCE="true" ENUM="false" NEXT="course"/>
```

Statements-elementti koostuu tietokantakomennon määrittelystä. Mikäli on tarpeen moduulia asennettaessa syöttää tietoja moduulin tietokantatauluihin, tehdään se kyseistä elementtiä käyttäen. Seuraavassa on esitetty moduulin asennuksen yhteydessä Moodlen oman lokitaulun päivittävä *statement*-elementti:

```
<STATEMENT NAME="insert_log_display" TYPE="insert" TABLE="log_display"
COMMENT="Initial insert of records on table log_display. Each record describes
how data will be showed by log reports.">
```

Tauluun päivitettävät tiedot määritetään *sentence*-elementissä määrittämällä moduuli, toimenpide, taulu sekä sarake:

```
<SENTENCE TEXT="(module, action, mtable, field) VALUES ('webcam', 'add',
'webcam', 'name')"/>
```

Videoaineistomoduurin toiminta edellytti kahden tietokantataulun määrittelyä ja asennusta Moodlen tietokantaan. Näistä ensimmäisen *webcam*-taulun sisältö perustuu moduulien yleiseen toimintaperiaatteeseen Moodlessa. Kyseistä taulua käytetään kursseihin liitettävien *webcam*-tyypin moduulien tietojen tallentamiseen. Taulukossa 5 on esitelty tauluun tallennettavat tiedot.

TAULUKKO 5. Webcam-taulun tietohakemisto

Webcam-taulu	
id	Instanssille automaattisesti määräytyvä ja perusavaimena toimiva järjestysluku
course	Sen kurssin <i>id</i> -luku, johon kyseinen instanssi on lisätty
name	Kurssinäkylässä näkyvä nimi aktiviteetille
intro	Aineiston lisääjän kirjoittama kuvaus aktiviteetista
introformat	Kuvauksen formaatti, joka tarkoittaa joko perustekstimuotoa tai HTML-muotoa
timecreated	Instanssin luontihetki aikaleimamuodossa
timemodified	Aikaleima instanssin edellisestä muokkaushetkestä

Webcam-instanssit sisältävän taulun lisäksi moduulin toiminta edellytti toista taulua, joka sisältää yksittäisiin instansseihin liitettävien videoiden tiedot. Tietyiltä osin myös tämän taulun määrittely on Moodlen yleisten periaatteiden sanelema. Lisättävät tiedot noudattavat suurelta osin samoja käytäntöjä kuin mitä Moodlen omissa kurssi- ja instanssitauluissa on sovellettu. Keskeisin poikkeavuus videomoduulin tapauksessa on tarve määrittää ne opiskelijat, joilla on oikeus nähdä tietty yksittäinen video. Tällä määrittelyllä mahdollistetaan henkilökohtaiseksi tarkoitetun palautteen anto opettajalta opiskelijalle. Taulukko 6 kuvaa videoiden tiedot sisältävän taulun rakenteen.

TAULUKKO 6. Webcam_video-taulun tietohakemisto

Webcam_video-taulu	
id	Videolle automaattisesti määräytyvä <i>id</i> -luku
title	Kurssinäkylässä näytettävä nimi videolle
description	Kuvaus videon sisällöstä
source	Videon lisääjä Moodlen <i>user</i> -taulusta saatavana <i>id</i> -lukuna
target	Niiden opiskelijoiden <i>id</i> -luvut, joilla on oikeus nähdä kyseinen videoaineisto
filename	Videon tiedostonimi, joka on tietoturvallisuus huomioiden muodostettu merkkijono
added	Videon lisäyshetki aikaleimana
webcamid	Sen <i>webcam</i> -instanssin <i>id</i> -luku, johon kyseinen video kuuluu

5.3 Tietokannan hallinta

Yhteensopivuuden säilyttämiseksi eri tietokantojen kanssa Moodlessa on olemassa joukko tietokannan hallinnassa käytettäviä funktioita. Moduulin tietokantakomentojen ja -hakujen toteutuksessa käytettiin *dmllib*-kirjaston olemassa olevia funktioita. Esimerkiksi uuden videon tietokantaan lisäävä funktio moduulin omassa kirjastossa hyödyntää Moodlen *insert_record*-funktia:

```
$save = new object();
$save->title = $name;
$save->description = $description;
$save->source = $USER->id;
$save->target = $target;
$save->filename = $filename;
$save->added = $added;
$save->webcamid = $webcamid;
insert_record("webcam_video",$save);
```

Esimerkissä on luotu uusi objekti *\$save*, jonka jäsenmuuttujat muodostuvat tietokantakomennon kohteena olevan taulun sarakkeiden nimistä. Koodissa käytetty objekti *\$USER* on yksi Moodlen globaaleista muuttujista, josta tässä tapauksessa saadaan selville sivustoa parhaillaan käyttävän käyttäjän *id*-tunnus. Tiedot tietokantaan lisäävälle funktiolle annetaan parametreina kohteena olevan taulun nimi sekä lisättävät tiedot sisältävä objekti. Vastaavasti kirjastoissa on omat funktionsa tietojen päivittämiseksi ja poistamiseksi.

Samaa Moodlen kirjastoa käyttäen toteutettiin myös tietokantahakujen luonti. Kirjastossa on olemassa useita erilaisia vaihtoehtoja tietokantakyselyjen toteuttamiseen. Tapauskohtaisen käyttötarkoituksen mukaan valikoidaan käytettävä funktio. Seuraavassa esimerkissä on toteutettu määrätyn tietokannassa olevan videon nimen ja tiedostonimen kysely:

```
$rs = get_recordset("webcam_video","id",$vid);
```

```

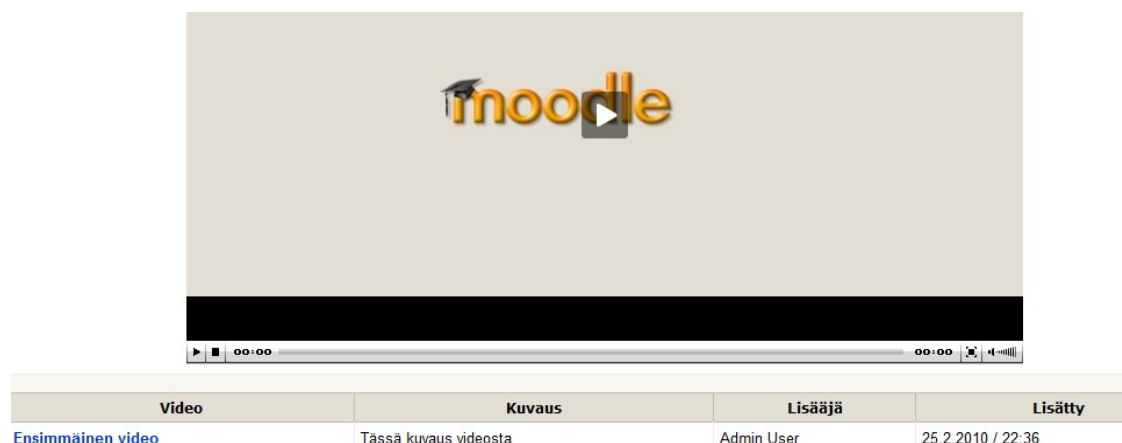
while ($videos = rs_fetch_next_record($rs)) {
    $title = $videos->title;
    $filename = $videos->filename;
}

```

Esimerkin haussa on haluttujen tietojen saamiseksi käytetty *get_recordset*-funktioita, jolle parametreina annetaan tässä tapauksessa haun kohteena olevan taulun nimi sekä taulusta tarkastettavan sarakkeen nimi ja sarakkeen arvo, joka esimerkissä on kahdesta merkkijonomuotoisesta parametrasta poiketen esitetty muuttujana. Edelleen samassa kirjastossa on määritelty funktio *rs_fetch_next_record*, jota käytetään haulla saatujen tietojen käsittelyyn. Esimerkin haussa tulokseksi on saatu vain yksi objekti nimeltään *\$videos*, joten silmukka ei tässä tapauksessa ole tarpeellinen. Kuitenkin mikäli tuloksen käsittelyssä oltaisiin tekemisissä useiden objektien kanssa, jäsenmuuttujien käsittely suoritettaisiin *while*-silmukassa.

5.4 Käyttöliittymä

Opiskelijan näkymä moduulista pyrittiin jättämään pelkistetyksi ja sitä kautta helppokäyttöiseksi. Tämä oli luonteva toteutusmalli, koska ainakaan tässä vaiheessa opiskelijoille ei ollut tarkoitusta tarjota muita toimintoja kuin videoiden katselu. Kuvassa 4 esitetään opiskelijalle näytettävä näkymä moduulista.



KUVA 4. Opiskelijanäkymä

Sivulle ensimmäiseksi latautuva video on viimeisin kyseisen opiskelijan katsottavaksi tarkoitetuista videoista. Video-objektin alapuolella näytetään listaus kaikista opiskelijalle tarjottavista videoista. Videon nimilinkki valitsemalla soittimeen ladataan valittu video. Videon nimen lisäksi listauksessa kerrotaan videon kuvaus, lisääjä ja lisäyksen ajankohta.

Kurssin opettajalle ensimmäisessä näkymässä näkyvät kaikki kurssin videot listattuna samalla tavoin kuin opiskelijoiden näkymässä. Tämän lisäksi näkyvissä on valintamahdollisuus opettajan lisäämien videoiden muokkaamiseen. Näkyvissä on luonnollisesti myös linkki uuden videon lisäämiseen. Linkistä siirrytään lomakenäkymään, johon videon lisääjä täyttää videon tiedot, jotka ovat:

- videon nimi
- opiskelijat, joilla on oikeus nähdä video
- kuvaus videosta

Lomakkeen ulkoasu on esitelty kuvassa 5. Mikäli kuvan kurssilla olisi rekisteröityjä opiskelijoita, näkyisivät heidän nimensä listauksessa. Opettajan on täten mahdollista kohdistaa video monivalintalistauksesta valitsemalla joko kaikille opiskelijoille, joukolle opiskelijoita tai yksittäiselle opiskelijalle.

Nimi	Kohde	Kuvaus
<input type="text" value="Toinen video"/>	<input type="text" value="Koko kurssi"/>	<input type="text" value="Lisättävä kuvaus"/>

KUVA 5. Videon tietojen määrittelylomake

Kuvassa näkyvää painiketta painamalla siirrytään videon tallentavalle sivulle, joka ei listauksen puuttumista lukuun ottamatta poikkea ulkoasultaan videon katselusivusta.

Moduulin ulkoisen olemuksen toteutuksessa hyödynnettiin *weblib*-kirjastosta löytyvää *print_table*-funktia. Funktiolle annettava parametri on taulun ominaisuuksista muodostettu olio. Olion keskeisimmät ominaisuudet vastaavat taulujen ominaisuuksia normaalissa HTML-kielessä. Olion ominaisuudet luetellaan seuraavassa:

- *\$table->head*
- *\$table->align*
- *\$table->size*
- *\$table->wrap*
- *\$table->data[]*
- *\$table->width*
- *\$table->tablealign*
- *\$table->cellpadding*
- *\$table->cellspacing*
- *\$table->class*
- *\$table->id*
- *\$table->rowclass[]*
- *\$table->summary*

Neljä ensimmäistä ominaisuutta määrittelevät taulun otsikot, kolumnien tasaukset ja koot sekä sen, miten teksti määritetään näkymään kolumnissa. Kukin edellä mainituista ominaisuuksista määritetään antamalla niille arvo taulukkomuodossa olevana listauksena. Tauluun sijoitettavan varsinaisen sisällön sisältävä *data[]*-ominaisuus poikkeaa edellä kuvatuista siinä, että ominaisuus on myös itsessään taulukko. Esimerkiksi taulun otsikkorivin arvot määritetään seuraavalla tavalla:

```
$video_table->head = array($strname,$strdescription);
```

Käyttökokemuksen helpottamiseksi Moodleen on kehitetty oma menetelmänsä valikkorakenteen toteuttamiseksi. Kuten muutkin ulkoasun toteuttamisessa käytettävät

funktiot on myös valikot tulostavat funktiot määritelty *weblib*-kirjastossa. Navigointi kurssinäkymässä on toteutettu oletuksena moduulin *view*-skriptissä seuraavasti:

```
$navlinks = array();
$navlinks[] = array('name' => $strwebcams, 'link' => "index.php?id=$course->id",
'< i>type' => 'activity');
$navlinks[] = array('name' => format_string($webcam->name), 'link' => "", 'type'
=> 'activityinstance');
$navigation = build_navigation($navlinks);
```

Skriptissä on luotu *\$navlinks*-niminen taulukko, jonka arvoiksi annetaan taulukkomuodossa sivunäkymässä näytettävä linkin nimi, linkin kohde sekä näkymän tyyppi. Muuttuja *\$navigation* välitetään tämän jälkeen sivunäkymän yläosan tulostavalle *print_header_simple*-funktiolle. Kuvassa 6 on esitetty esimerkin mukaisessa tapauksessa tulostuva valikko.



KUVA 6. Navigaatiovalikko

Koska toteutettu moduuli jakautui rakenteeltaan useampaan eri sivunäkymään, piti navigaatioon lisätä tasoja, jotta esimerkiksi lomakenäkymästä voidaan siirtyä kätevästi takaisin listausnäköön. Tämä tapahtui toistamalla esimerkissä esitetyn taulukon saama toinen arvo asianmukaisin parametrein sekä muuttamalla listausnäkötaulukon linkin kohteen määrittävän parametrin arvo. Funktion *print_header_simple* toimintaa ei siis ollut tarvetta määritellä uudestaan.

5.5 Kieliversioiden hallinta

Kieliversiot voidaan Moodlessa tallentaa käytettäväksi kahdella eri tavalla. Asennuksen jälkeen kaikki kielimäärittelyt sijaitsevat päähakemiston *lang*-kansiossa. Kansiossa eri kielet on jaettu omiin kansioihinsa, jotka on nimetty niin, että kansion nimessä on määritelty kyseessä olevan kielen lyhenne sekä alaviivalla erotettuna käytetty merkistökoodaus, jolloin esimerkiksi UTF-8 -koodatut suomenkieliset kielimäärittelyt löytyvät kansiota *fi_utf8*. Moduulien ja lohkojen asennuksen helpottamiseksi Moodlessa on kuitenkin olemassa ominaisuus, joka löytää kielimäärittelytiedostot myös asennettujen elementtien omista hakemistoista. Tällöin moduulien omista hakemistoista löytyvän edellä kuvatun kaltaisen *lang*-kansion sisältöä ei ole tarpeen erikseen siirtää varsinaiseen kielimäärittelyhakemistoon. Kieliyhteensopivuuden takia *string*-muuttujat määritellään koodissa seuraavalla tavalla:

```
$strwebcam = get_string("modulename", "webcam");
```

Funktio *get_string* on *moodlelib*-kirjastossa määritelty funktio, jolle tässä tapauksessa annetaan parametreina halutun *string*-muuttujan nimi sekä sen tiedoston nimi, josta määrittelyt löytyvä. Moodlen ohjelmakoodissa *string*-tyypin muuttujat nimetään *str*-alkuisiksi muuttujien hahmottamisen helpottamiseksi. Yllämainittuun *\$strwebcam*-muuttujaan käytetty funktio palauttaa *webcam.php*-tiedostosta löytyvän arvon. *Webcam*-skriptissä kuten muissakin kielimäärittelytiedostoissa *string*-muuttujien erikieliset versiot esitetään seuraavanlaisesti:

```
$string['modulename'] = 'Webcam';
```

Skriptissä jokainen rivi koostuu siis *string*-tyyppisen muuttujan nimestä sekä tälle annettavasta arvosta. Suomenkielinen arvo määriteltäisiin näin ollen suomenkielisessä skriptissä seuraavasti:

```
$string['modulename'] = 'Nettikamera';
```

Mikäli määriteltävä arvo sisältää esimerkiksi skandinaavista merkistöä, tulee nämä merkit esittää tarpeen vaatiessa UTF-8 -koodattuina.

6 POHDINTA

Työn lähtökohtana oli virtuaaliseen oppimisympäristöön yhdistettävä uudenlainen opetuksellinen elementti, joka mahdollistaisi palautteen antamisen yksinkertaisemmin web-kameran välityksellä kirjallisen esitysmuodon sijaan. Työn tarkoituksena ei ollut niinkään Moodlen opetuksellisten ominaisuuksien esittely käytännössä kuin uudenlaisten oppimismetodien hyödyntämisen liittäminen jo käytössä olevaan oppimisalustaan. Moodlen tehokkaasta käytöstä on kirjoitettu useita käytännönläheisiä oppaita sekä opettajan että opiskelijan näkökulmasta. Tässä työssä tarkasteltiin sekä tarpeita Moodlen kehittämiseen paremmin opetuksessa hyödynnettäväksi että yleisemmin virtuaaliopiskelun mukanaan tuomia mahdollisuuksia.

Vaihtoehtoisia menetelmiä moduulin toteuttamiseen oli useita lähtien eri palvelin- ja soittinratkaisuista. Wowza Media Server valikoitui käytettäväksi mediapalvelimeksi sekä toimivuutensa että lisensoinnin vuoksi. Vapaan lähdekoodin Red5-palvelimen pitäisi ainakin teoriassa vastata toiminnaltaan Wowzan palvelinta, mutta esimerkiksi dokumentaatiossa on vielä Wowzaan verrattuna kehitettävää. Sen tarkemmin Red5-palvelimen toimivuutta opinnäytetyön aiheen mukaisessa käytössä ei prosessin aikana tarkasteltu, koska valinta oli vapaasti tehtävissä eri vaihtoehtojen välillä.

Tapauskohtaisesti voi kuitenkin olla niin, että ilmainen Red5 koetaan houkuttelevammaksi vaihtoehdoksi. Hyvin todennäköisesti ilmaiseen vaihtoehtoon pätee myös tässä yhteydessä samat periaatteet kuin verrattaessa Moodlea kaupallisiin oppimisalustoihin. Wowza on käytännössä kaikilta ominaisuuksiltaan sellaisenaan käytettävissä, kun taas Red5-palvelimen käyttöönotto edellyttää tarkempaa perehtymistä palvelimen toimintaan ja dokumentaatioon. Toisaalta vapaa lähdekoodi mahdollistaa myös palvelimen muokkauksen käyttötarkoituksen asettamien vaatimusten mukaiseksi.

JW Playerin valikoitumiseen käytettäväksi Flash-soittimeksi pätevät samat seikat kuin mediapalvelimen valinnassa. JW Player oli toimiva ja vapaasti käytettävissä oleva vaihtoehto. Epäkaupallisessa käytössä JW Player on lisäksi maksuton. Lopullisen käyttötarkoituksen mukaan onkin selvitettävä, onko valittava maksullinen versio vai

jokin vastaava maksuton vaihtoehto. Työn aikana ei sen tarkemmin selvitetty myöskään eri soitinvaihtoehtojen välisiä eroja jo pelkästään eri vaihtoehtojen laajan kirjon vuoksi. Selväksi kuitenkin kävi, että eri soitinten ominaisuudet ja toimintaperiaatteet vastaavat suurelta osin toisiaan, joten työn toteuttamisessa käytetyt menetelmät ovat suurella todennäköisyydellä ainakin sovellettavissa myös muita Flash-soittimia käytettäessä.

Moduulin toiminta suunniteltiin ja toteutettiin käytettävyys, helppokäyttöisyys ja loogisuus huomioiden. Moduuli käyttöönotettaessa on testattava sen toiminta käytännössä oppilaitoksen omassa Moodle-oppimisalustassa. Koska kyseessä on dokumentaation pohjalta kehitetty lisäosa, on varmistuttava moduulin toimivuudesta suorittamalla järjestelmällinen testaus ennen käyttöönottoa. Työn yhteydessä testausta ei ollut mahdollista suorittaa. Ilman käytännön kokemuksia on myös mahdotonta olla täysin selvillä mahdollisista muutostarpeista esimerkiksi käyttökokemuksen luontevuuden osalta.

LÄHTEET

Armitage, Amy 2009. Streaming without a streaming server with Flash (pseudo streaming). WWW-dokumentti. http://www.articlealley.com/article_1161605_2.html. Luettu 21.2.2010.

Colburn, Mike 2010a. Difference between progressive (HTTP) delivery and streaming. WWW-dokumentti. <http://www.rentanencoder.com/archives/221>. Luettu 17.2.2010.

Colburn, Mike 2010b. What is a Flash player – Do I need one?. WWW-dokumentti. <http://www.rentanencoder.com/archives/218>. Luettu 17.2.2010.

Coldewey, Devin 2009. HTML 5: Ogg Theora vs H.264 in the battle for a web video standard. WWW-dokumentti. <http://techcrunch.com/2009/07/06/html-5-ogg-theora-vs-h264-in-the-battle-for-a-web-video-standard/>. Luettu 24.2.2010.

Flinkman, Ritva, Heikkinen, Eija, Mikkonen, Liisa, Mustonen, Raimo, Niiranen, Esa, Ruotsalainen, Mervi & Simanainen, Raili 2006. Kokemuksia Kajaanin ammattikorkeakoulun verkko-opetuksen kehittämishankkeista. Kajaani: Kajaanin ammattikorkeakoulu.

Heynderickx, Jim 2006. Moodle: A Question of Change. WWW-dokumentti. <http://www.k12converge.com/?p=40>. Päivitetty 21.1.2009. Luettu 21.1.2009.

Kalliala, Eija 2002, Verkko-opettamisen käsikirja. Jyväskylä: Gummerus Kirjapaino Oy.

Karevaara, Samuli 2009. Moodlen perusteet – Opettajan ja opiskelijan opas. Tampere: Esa Print Oy.

Keränen, Vesa & Penttinen, Jukka 2007. Verkko-oppimateriaalin tuottajan opas. Porvoo: WSOYpro/Docendo-tuotteet.

Knowles, Jesse 2007. Streaming video in Flash. WWW-dokumentti. <http://www.jesseknowles.com/index.php/flash-development/streaming-video-in-flash>. Luettu 17.2.2010.

Manninen, Pasi & Marttila, Jarno 2006. Flash 8 & ActionSript. Porvoo:Docendo Finland Oy.

Meisalo, Veijo, Sutinen, Erkki & Tarhio, Jorma 2000. Modernit oppimisympäristöt. Juva: Tietosanoma Oy.

Meisalo, Veijo, Sutinen, Erkki & Tarhio, Jorma 2003. Modernit oppimisympäristöt. Pieksämäki: RT-Print Oy.

Norvanto, Tuomo 1998. Videoneuvottelu opetuksessa. Turku: Painosalama Oy.

Paananen, Petteri 2008. Flash-julkaisijan opas. Saarijärvi: WSOYpro/Docendo-tuotteet.

Rice IV, William H. 2006. Moodle – E-Learning Course Development. Birmingham: Packt Publishing Ltd.

Silander, Pasi & Koli, Hanne 2003. Verkko-opetuksen työkalupakki – oppimisaihioista oppimisprosessiin. Saarijärvi: Saarijärven Offset Oy.

Vahtivuori, Sanna 2001. Kohti yhteisöllisen ja kokemuksellisen verkko-opetuksen suunnittelua – käyttäjät suunnittelun polttopisteessä. Teoksessa Tella, Seppo, Nurminen, Outi, Oksanen, Ulla & Vahtivuori, Sanna (toim.). Verkko-opetuksen teoriaa ja käytäntöä. Helsinki: Hakapaino.

Yli-Luoma, Pertti V. J. & Pirkkalainen, Lauri 2005. Verkko-oppimisen työvälineitä. Saarijärvi: Saarijärven Offset Oy.